

An efficient snow flake schema with hash map using SHA-256 based on data masking for securing employee data

Tumkur Shankaregowda Bharath, Channakrishnaraju

Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumkur, India

Article Info

Article history:

Received May 30, 2024

Revised Oct 4, 2024

Accepted Oct 17, 2024

Keywords:

Data confidentiality

Data masking

Sensitive data

SHA-256

Snow flake schema

ABSTRACT

In various organizations and enterprises, data masking is used to store sensitive data efficiently and securely. The data encryption and secret-sharing-based data deploying strategies secure privacy of subtle attributes but not secrecy. To solve this problem, the novel snowflake schema with the hash map using secure hash algorithm-256 (SHA-256) is proposed for the data masking. SHA-256 approach combines data masking by secret sharing for relational databases to secure both privacy as well as the confidentiality of secret employee data. The data masking approach supports preserving and protecting the privacy of sensitive and complex employee data. The data masking is developed on selected database fields to cover the sensitive data in the set of query outcomes. The proposed method embeds one or multiple secret attributes about multiple cover attributes in a similar relational database. The proposed method is validated through different performance metrics such as peak signal-to-noise ratio (PSNR) and error rate (ER) and it achieves the values of 50.084dB and 0.0281 when compared to the existing methods like Huffman-based lossless image coding and quad-tree partitioning and integer wavelet transform (IWT).

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Tumkur Shankaregowda Bharath

Department of Computer Science and Engineering, Sri Siddhartha Institute of Technology

Maraluru, SSAHE, Tumkur, India

Email: bharathts@ssit.edu.in

1. INTRODUCTION

Reversible data hiding (RDH) is an innovative approach, which keeps the cover data secret and permits the data hider to insert more data in it [1]. The data can be hidden with any covered digital media data including image, text, video as well as audio [2]. Data sharing or data encryption is the most significant method for protecting the confidentiality of outsourced databases [3]. Significant data-hiding applications consist of secret data storage as well secret communication (steganography), copyright cover media protection as well as media verification. The data-hiding strategy is divided into various parts such as reversible and irreversible. The reversible data-hiding strategy permits a media to entirely recovered after an extraction of secret data [4], [5]. Encryption of an image denotes the conversion of the actual image into a confused image by a specific approach. An integration of image encryption and RDH technology is known as RDH of encrypted images (RDHEI) [6], [7]. This approach is segmented into multiple divisions such as content owner, data hider as well as receiver [8]. Initially, a content owner encrypts the data and it is forwarded to a data hider. Then, secret data can be masked by a data hider without knowledge of the original data. Eventually, accurate data extraction as well as efficient data recovery can be attained only when the receiver has both encryption and decryption keys [9], [10]. RDHEI approach comprehends firm communication and reversibility, hence, it is broadly utilized in various areas like judicial domains, medical

imaging, and military images [11], [12]. The existing researchers developed the RDHEI algorithms for data masking and they can be approximately divided into three types such as vacating rooms before encryption, vacating rooms in encryption and vacating rooms after encryption [13], [14]. Even though the researchers have introduced the various RDHEI approaches, however some issues need to be addressed like an incapability to attain a greater embedding rate and an incapability to significantly extract all the data [15]. To solve these problems, this research effectively proposed the snowflake schema algorithm with hash mapping using secure hash algorithm-256 (SHA-256).

Su and Chang [16] developed the RDHEI plan which utilized a Huffman-based lossless image coding approach to obtain the embedding dimensions. The large embedding was designed due to the maximum compression rate as well as the coding efficiency of an integrated plan. The suggested approach aimed to set up a private as well as secure communication channel among sender and receiver through data embedding into encrypted images. The developed method also ensured the error-free extraction of privacy messages. However, the bit extraction defeated risk and image recovery gained on the architectural extent whose spatial correlation was poor. Wang *et al.* [17] implemented a novel method for privacy-preserving reversible data masking. The technique utilized quad-tree partitioning and integer wavelet transform (IWT) techniques. Firstly, the encrypted image was transformed using IWT, and the coefficients in high-frequency subbands were converted into an 8-bit binary representation. Subsequently, a quad-tree partitioning approach was employed to encode every coefficient block. This approach yielded a significant improvement in the embedding rate while ensuring lossless recovery of an actual image across various embedding rates. Moreover, the extracted data exhibited no errors. However, the statistical features of encrypted and embedded images were complex for detecting the data. Zhang *et al.* [18] presented the high-capacity RDHEI according to block-wise multi-predictor as well as enhanced Huffman coding for encrypted images. Initially, the actual image was categorized into non-overlapping blocks, and 16 prediction approaches were utilized for the prediction of current pixel value from nearest pixels. The enhanced Huffman encoding was utilized to wrap the forecast error image as well as optimal prediction approach was identified through total prediction errors. The Huffman code had easily encoded and decoded the data but it required the frequency of every symbol to be known in prior.

Rahmani *et al.* [19] implemented a new scheme that integrated the data hiding through secret sharing for relational databases to secure secret attributes. The suggested schema had embedded various secret characteristics in the relation into various concealment characteristics in a similar relational database. The group of columns was developed such that attributes were pretended to correlate with the cover attributes. However, an index column must be developed for every secret attribute to share columns which was similar to the meta-data-based SSDO schemas. Anushiadevi *et al.* [20] presented the RDH for an encrypted image by utilizing a most significant bit (MSB) variance of the pixel value. By this approach, the third parties could embed the ciphertext in a cipher image without knowledge of a secret as well as privacy data. The person with the decryption key could return the privacy as well as the envelope without any information loss. The suggested approach recovered the images secretly without the loss when the channel was noise-free. However, the image cover as well as the secret wasn't correctly retrieved due to the presence of noise. Malik *et al.* [21] introduced the data-hiding approach in encrypted text named RDHET. Primarily, an actual data was changed to ASCII values and then, the Paillier cryptosystem was embraced to encrypt the text data and transform it to a data hider for the following process. Secret data was entrenched into homomorphically encrypted text by the approach which didn't lose any data called the Paillier cryptosystem. Eventually, an entrenched privacy data as well as the original data were recovered at the receiving side. The suggested approach acquired the RDH functionality in an encrypted domain (RDHED), however, an approach transcended for a text-based ED. Murthy and Manikandan [22] implemented the RDH scheme that considered the block-wise image histograms. In this implemented approach, the meta-data needed for a block was embedded within a similar block in such a way that the receiver could employ the image recovery as well as extraction of data. In the process of implemented data hiding, the implemented approach utilized the marker data to differentiate among the blocks. The suggested approach consumed much time for the encryption procedure and lack of process in real-time applications.

The highlights of the proposed method are listed in the following:

- This research proposes an efficient data masking approach for relational databases according to the snowflake schema with hash map using SHA-256 in the office environment.
- The non-sensitive attributes have been selected as semi-cover attributes to describe virtual share columns. Data hiding process develops some original share columns combined with cover attributes.
- The effectiveness of the proposed method is validated by different evaluation metrics like execution time, peak signal-to-noise ratio (PSNR), and error rate (ER).

This research is presented as follows: section 2 defines the proposed method. Section 3 illustrates the encryption using snowflake schema. Section 4 provides the results and discussion and section 5 provides the conclusion of the research.

2. PROPOSED METHOD

In this research, the novel snowflake schema is proposed for securing employee data. Data masking as well as secret masking are efficient methods for protecting the both privacy and confidentiality of data. The proposed schema uses the redundancy to embed the data for RDH. Every relation consists of one or more secret attributes and which are selected as the concealment attributes. The non-sensitive attributes have been selected as semi-cover attributes to describe virtual share columns. Simultaneously, a developed original share columns through the described virtual share columns are utilized to recover the secret attributes. The objective of this research is to design a scheme for identifying sensitive data and content format preserving procedure on the masked data by developing a technique to retain the size of the masked data as same as the original data. Figure 1 illustrates the architecture of the proposed method. Simultaneously, the developed original share columns with described virtual share columns are utilized to recover secret attribute (s).

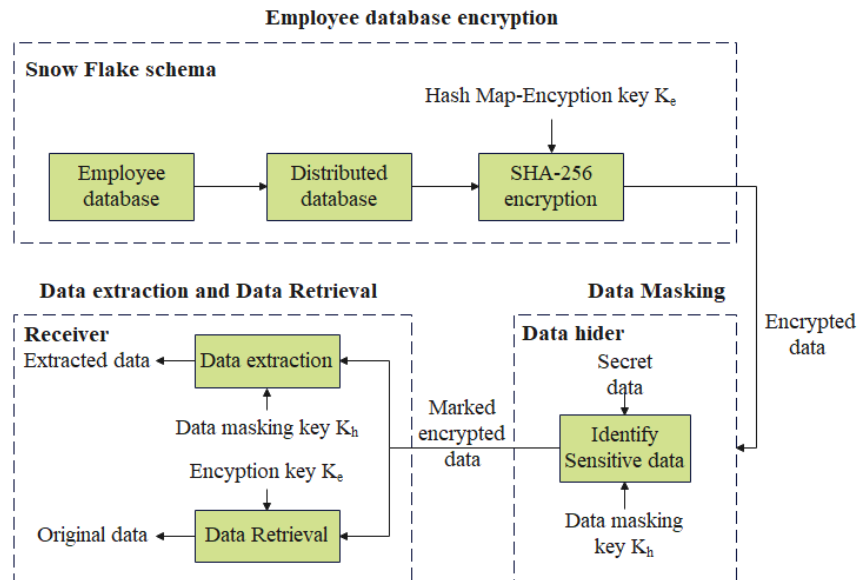


Figure 1. Framework of the proposed method

2.1. Employee database

In this section, an employee data plays a significant role for enhancing the employing facilities by utilizing data analytics from different sources. The primary aim of employee data is to secure the employee personal information. The sensitive data in employee's personal data refers to employee ID, employee name, email ID, social security administration (SSA) number, credit card number, and date of birth. These data can be stored on the relational database named snowflake schema and the information about this schema is discussed in next section.

3. ENCRYPTION USING SNOW FLAKE SCHEMA

The snowflake schema in data masking is used for securing the employee data features that permits the organizations to mask sensitive data in their database tables, views, and query results in the real-time. The schema is used for protecting the sensitive information from unauthorized access or vulnerability, it supports sensitive data in organizations without the need for manual data modifications or multiple copies of databases. It simplifies data privacy and security by dynamically applying data masking rules according to user roles, ensuring that only authorized users can access sensitive information while maintaining the usability of the data for other users. Figure 2 shows the access level control of snowflake schema.

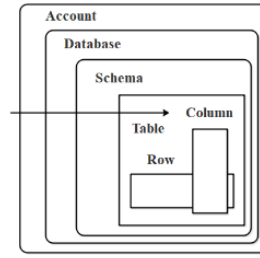


Figure 2. Access level control of snowflake schema

The snowflake schema allows constituting hierarchies utilizing various sub-dimensions which are compact as well as normalized relational tables. Therefore, snowflake schema is proposed according to the column databases. This schema primarily retains an initial two transformation guidelines and after append the two various guidelines which permits the constituting orders in column databases as:

Transformation 1: every dimension $D_i (Name^{D_i}, A^{D_i}, H^{D_i}) \in D^{MDM}$ is transferred into the group of dimensional nodes described by $(Name^{D_i^{MGD}}, A^{D_i^{MGD}}, H^{D_i^{MGD}})$ where:

- $Name^{D_i^{MGD}}$ is dimension D_i name is integrated with function λ^{MGD} as the label to a dimension nodes N_{D_i} .
- $A^{D_i^{MGD}}$ is set of parameters as well as weak dimension D_i attributes. Every constraint is transformed into group of nodes to permit the constituting orders. Every base parameter feature is changed into constraint nodes.
- $H^{D_i^{MGD}}$ is set of nodes constituting hierarchies of the dimension D_i .

Transformation 2: hierarchies $(Name^{H_j}, Param^{H_j}, Weak^{H_j})$ are transformed into the set of linked nodes $(Param^{H_j^{MGD}}, Weak^{H_j^{MGD}})$ where:

- $Param^{H_j^{MGD}}$ is set of parameters of parameters nodes. The function λ^{MGD} integrates to these constraint nodes $Param^{H_j}$ as the label.
- $Weak^{H_j^{MGD}}$ is set of properties that integrated to the constraint nodes utilizing a function σ^{MGD} .
- An edge is described among the fact nodes N_{F_i} as well as a minimal associated parameter $Param_k$ of every dimension utilizing the function ρ^{MGD} .
- The edges are described among the neighboring parameters of the similar hierarch utilizing a ρ^{MGD} function.

3.1. Hash function

In the hash function [23], both the input and output data values are alphanumeric. The arithmetic function exchanges an input value or arbitrary data length into an output value. A value returned by a hash function is known as a message digest or hash value. The famous hash functions contain some required properties named deterministic, unique, one-way, and uncorrelated. The process by which the hash function converts the message with a finite length into one with a predefined length. Encryption is the procedure of masking the message content from the outsource.

3.1.1. Secure hash algorithm-256

By utilizing a SHA-256 approach, the employee data to be encrypted. The SHA-256 [24], [25] is one of the cryptographic hash algorithms. The set of data signatures is similar to the cryptographic hash functions. The SHA is a hash function, while the SHA-256 attains the best values as compared to the two bits of original data. The approach will distribute the various hash outcomes even if one symbol has been modified. The hash functions are outstanding for the integrity of data checks, challenge hash authentication, digital signature, and blockchain applications as well as anti-tamper since it is a one-way function. The SHA designs the substantially distinct fixed length $n = 256$ bits (32-bytes) hash value. SHA-256 carries the input message M of a maximum length of 256 bits. M is continuously padded through p-bit according to design input of length L , a product of 512 where the final 64 bits are taken for 232 of the raw input data. The padded bits begin with 1 which is subsequent through the essential numbers of 0. Thus, a last input to SHA-256 becomes the length of $L = 1 + p + 64$, which is divided into $N = L/512$ blocks of 512 bits. SHA-256 begins with the 8 32-bits which are predetermined by the initial vectors $H_0 = h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7$ which is updated every 64 following rounds and eventually generates a hash value H_t . Figure 3 shows the entire procedure of SHA-256 algorithm.

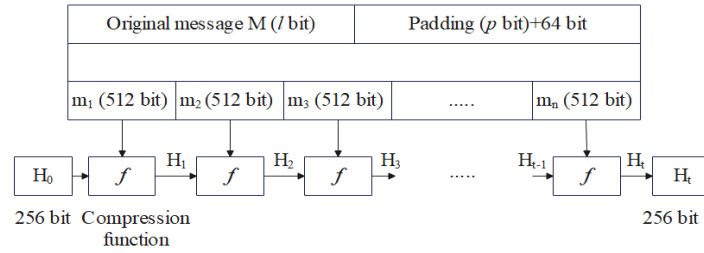


Figure 3. Hash value generation in SHA-256

3.2. Encrypted data

At the side of content owner, an actual data is transferred into ASCII values, after that every value is encrypted by utilizing Paillier cryptosystem using public key (N, g) , where $N = p \times q$ is a product of two prime numbers and $g \in Z_{N^2}^*$ which intersects the condition $gcd\left(\frac{g^\lambda \text{mod } N^2 - 1}{N}, N\right) = 1$, where λ is denoted as the secret key, which subsequent a state $\lambda = lcm(p - 1, q - 1)$. Assuming actual data as T , with size L , the ASCII value of every part in a range among $[0, 255]$ and represented through 8 bits. By using (1), a data owner of an actual data encrypts every ASCII value of T :

$$T_c(i) = E[T(i), r] = g^{T(i)} r^N \text{mod } N^2 \tag{1}$$

where, $T_c(i)$ is ciphertext of every original data value, where i in the range of $1 \leq i \leq L$. The r is arbitrarily chosen for every ASCII value to attain semantic security. Furthermore, $E[\cdot]$ depicts an encryption function. Eventually, encrypted data is depicted as T , which is utilized through a data hider to add more secret data S . After the encryption of employee data, the encrypted data are forwarded to the process of data masking.

3.3. Data masking

After acquiring encrypted data T , a data masker is capable of efficiently embedding an extended secret data S into T , which results in any information loss. The data masker is not required to be careful about original data corresponding to embedding a secret data S . This is established through using homomorphic as well as probabilistic possessions of the Paillier cryptosystem, which permits lossless data masking as well as the generation of marked encrypted data. The secret data is embedded with an 8 mn length of flattened data. Hence, data masking embeds secret data in final process of encrypted data based on the compressed data length. To assure privacy, secret data is encrypted before embedding a data. A pseudorandom stream is developed by the data-hiding key K_h , and a bitwise XOR operation among secret data as well as a pseudorandom stream is estimated to acquire a sequence of encrypted secret data. The bits in encrypted data without flattened data can be masked to cover the data. In this research, the proposed method has embedded encrypted secret data by bit replacement of K_h and eventually, the marked encrypted data is acquired. Eventually, the masked data are then provided for both data extraction and data retrieval.

3.4. Secret data extraction and data retrieval

The secret data extraction, as well as data retrieval, are the two conditions performed on receiver side. The secret data is extracted through the data hider by utilizing a shared crawl key τ . Primarily, the secret data is encrypted, and after it is embedded into encrypted data by utilizing a crawl key τ . After extraction, τ is used to regenerate actual secret data. An extraction of secret data is formulated in (2). In (3) depicts the decryption procedure to retrieve actual data:

$$S_k = \begin{cases} 0, & \text{if } T_{me}(i) \text{mod } 2 = 0 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

$$R(i) = D[T_{me}(i)] = \frac{L(T_{me}(i)^\lambda) \text{mod } N^2}{L(g^\lambda \text{mod } N^2)} \text{mod } N \tag{3}$$

where $R(i)$ is directly decrypted data having the constraints $L(x) = (x - 1)/N$ and $D[\cdot]$ depicts a decryption function. After decryption, the last retrieved actual data R is similar to actual data after exchanging into their character values T . The actual data is employed in both the constraints of encrypted data and marked data by utilizing a secret key (λ) .

4. RESULTS AND DISCUSSION

The efficiency of the proposed method is executed according to the simulation results. This simulation result gives effective modelling and simulation over the data masking services. Table 1 shows the hardware as well as the simulation settings of the proposed method. The efficiency of the SHA-156 method is validated through different performance metrics such as PSNR, ER, encryption time, decryption time, computational time, and execution time which is formulated in (4) to (9):

$$PSNR = \log_{10} \frac{S_{max}^2}{MSE} \tag{4}$$

$$ER = \frac{\text{Number of error bits}}{\text{Total number of bits}} \times 100 \tag{5}$$

$$\text{Encryption Time} = En(x) = (x + n) \bmod 26 \tag{6}$$

$$\text{Decryption Time} = Dn(x) = (x - n) \bmod 26 \tag{7}$$

$$\text{Computational Time} = \text{Instruction count} * \frac{CPI}{\text{Clock rate}} \tag{8}$$

$$\text{Execution Time} = \text{Clock cycle time} \times \text{No. of instrs} \times \text{avg CPI} \tag{9}$$

Table 1. Hardware specification of the proposed method

Required	Specifications of the components
Environment	Python 3.8
Processor	Intel core i5 processor
Operating system	Windows 10 64-bit OS
Hard disk	1 TB
RAM	16 GB
System	64 Bit OS

4.1. Performance analysis

The effectiveness of a proposed SHA-256 based data masking process is validated with the previous data masking approaches. The proposed method’s performance is validated by different performance metrics like encryption time, decryption time, computational time, retrieval time, execution time, PSNR, and ER. Table 2, Figures 4, and 5 denote the performance evaluation of encryption time and decryption time. Table 3 and Figure 6 to 8 denote the performance evaluation of computational time, retrieval time, and execution time. Table 4 and Figure 9 denote the performance analysis of PSNR. Table 5 and Figure 10 denote the performance evaluation of ER.

Table 2. Performance evaluation of encryption and decryption time

Methods	Encryption time (ms)					Decryption time (ms)				
	Data size					Data size				
	200	400	600	800	1,000	200	400	600	800	1,000
RSA	1,086	1,286	1,428	1,459	1,574	930	1,089	1,100	1,167	1,189
AES	986	1,186	1,356	1,295	1,389	1,020	974	968	1,084	1,018
MD5	956	974	1,248	1,056	1,269	968	869	978	994	956
Blow fish	853	960	1,078	1,128	1,210	786	846	556	989	878
SHA-256	713	780	893	983	990	630	758	789	859	841

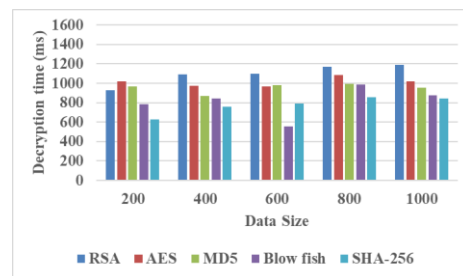
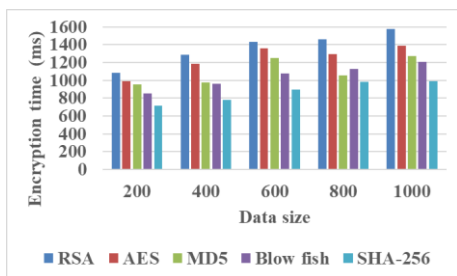


Figure 4. Graphical representation of encryption time Figure 5. Graphical representation of decryption time

Table 3. Performance evaluation of computational, retrieval and execution time

Methods	Computational time (ms)					Retrieval time (ms)					Execution time (ms)				
	Data size					Data size					Data size				
	200	400	600	800	1,000	200	400	600	800	1,000	200	400	600	800	1,000
RSA	589	568	425	598	625	289	325	310	324	356	865	925	1,020	1,100	1,220
AES	614	785	798	857	894	323	348	378	459	484	868	872	969	989	1,020
MD5	849	869	898	987	914	378	357	464	559	593	1,100	1,198	1,199	1,274	1,464
Blowfish	560	589	623	689	712	289	325	389	395	425	789	800	825	854	925
SHA-256	464	355	365	378	464	228	223	278	439	456	798	852	849	879	988

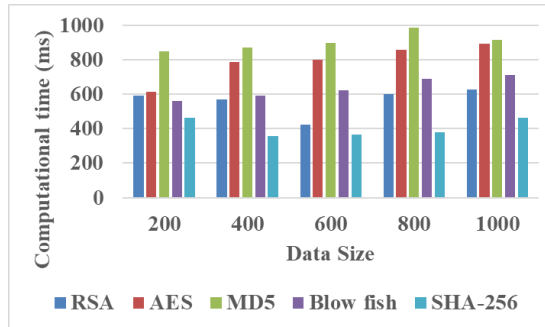


Figure 6. Graphical representation of computational time

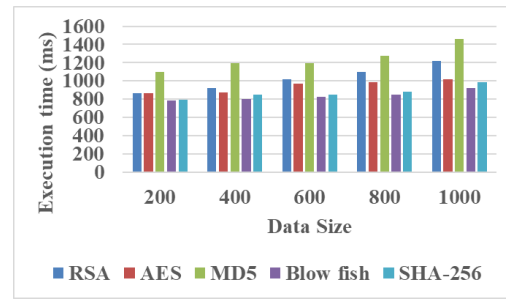
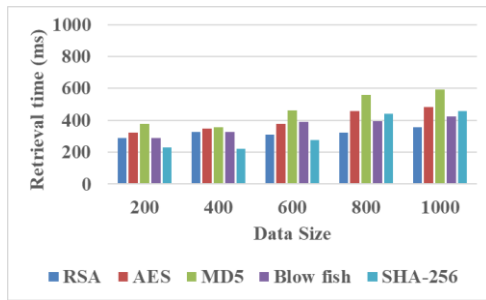


Figure 7. Graphical representation of retrieval time

Figure 8. Graphical representation of execution time

Table 4. Performance evaluation of PSNR (dB)

Methods	Data size				
	200	400	600	800	1,000
RSA	51.025	52.25	52.96	53.06	53.86
AES	50.084	49.248	50.055	50.524	51.194
MD5	53.094	52.148	51.055	52.080	53.194
Blowfish	59.250	58.98	59.454	60.250	60.783
SHA-256	49.024	50.068	49.145	49.270	50.084

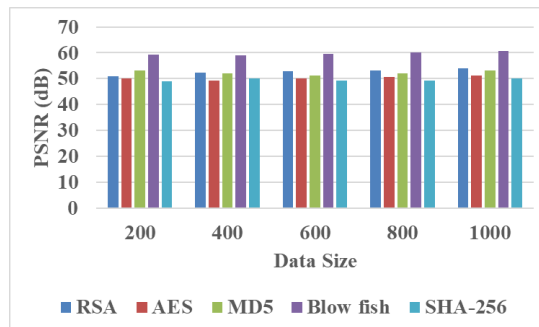


Figure 9. Graphical representation of PSNR

Table 2, Figures 4, and 5 demonstrates the performance evaluation of SHA-256 method using encryption and decryption time. The different data sizes such as 200, 400, 600, 800, and 1,000 have been taken during the evaluation process. The proposed SHA-256 performance is validated with previous encryption algorithms like Rivest Shamir Adleman (RSA), advanced encryption standard (AES), and blowfish approach. The proposed SHA-256 achieves encryption times of 713, 780, 893, 983, and 990 at data sizes of 200, 400, 600, 800, and 1,000 respectively. The proposed SHA-256 achieves the decryption time of 630, 758, 789, 859, and 841 at data sizes of 200, 400, 600, 800, and 1,000 correspondingly.

Table 3 and Figures 6 to 8 illustrates the performance evaluation of the SHA-256 method using computational time, retrieval time and execution time. The different data sizes such as 200, 400, 600, 800, and 1,000 have been considered during the evaluation process. The proposed SHA-256 effectiveness is estimated with the existing methods like RSA, AES, and blowfish. The proposed SHA-256 achieves the computational time of 464, 355, 365, 378 and 464. The proposed SHA-256 achieves the retrieval time of 228, 223, 278, 439, and 456. The proposed SHA-256 achieves the execution time of 798, 852, 849, 879, and 988 at the data sizes of 200, 400, 600, 800, and 1,000 correspondingly.

Table 4 and Figure 9 depict the performance evaluation of the SHA-256 method in terms of PSNR. The different data sizes such as 200, 400, 600, 800, and 1,000 have been considered during the evaluation process. SHA-256 performance is validated with the previous encryption approaches like RSA, AES, and blowfish. The proposed SHA-256 achieves the PSNR of 49.024, 50.068, 49.145, 49.270, and 50.084 at the data sizes of 200, 400, 600, 800, and 1,000 correspondingly.

Table 5. Performance evaluation of ER

Methods	Data size				
	200	400	600	800	1,000
RSA	0.0225	0.0325	0.0356	0.0389	0.0423
AES	0.0221	0.0236	0.0303	0.0313	0.0340
MD5	0.0440	0.0413	0.0403	0.0421	0.0336
Blowfish	0.0356	0.0450	0.0556	0.0504	0.058
SHA-256	0.0100	0.0113	0.0148	0.0236	0.0281

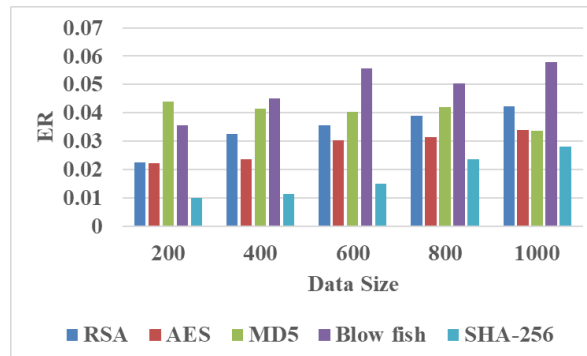


Figure 10. Graphical representation of ER

Table 5 and Figure 10 denote the performance evaluation of the SHA-256 method in terms of ER. The different data sizes such as 200, 400, 600, 800, and 1,000 have been taken for evaluating the performance of the SHA-256 method. The SHA-256 effectiveness is estimated with previous approaches like RSA, AES, and blowfish. The proposed SHA-256 achieves the ER of 0.0100, 0.0113, 0.0148, 0.0236, and 0.0281 at the data sizes of 200, 400, 600, 800, and 1,000 correspondingly.

4.2. Comparative analysis

Table 6 portrays the comparison of the proposed method with previous approaches. The proposed method is assessed by different performance metrics like encryption time, decryption time, retrieval time, PSNR, and ER. The existing methods such as [16]-[18], [22] are compared with the proposed method to validate the effectiveness. The existing method's values are simulated according to the proposed method values.

Table 6. Comparative of proposed method with previous works

Methods	Encryption time (ms)	Decryption time (ms)	Retrieval time (ms)	PSNR (dB)	ER
Huffman-based lossless image coding [16]	1492	903	567	80.0863	1.6488
Quad-tree partitioning and IWT [17]	1345	922	534	60.3986	1.5679
Enhanced Huffman coding [18]	1123	901	502	55.8974	4.1588
RDH scheme with block-wise image histograms [22]	1001	889	489	50.9405	0.0210
Proposed method (SHA-256)	990	841	456	50.084	0.0281

4.3. Discussion

The limitations of previous approaches and advantages of the SHA-256 approach are discussed in this section. The Huffman-based lossless image coding [16] had the limitation in poor bit extraction and image recovery. Quad-tree partitioning and IWT [17] had the analytical features of encrypted and embedded encrypted images that were difficult to detect. Enhanced Huffman coding [18] had easily encoded and decoded the data but it required the frequency of every symbol to be known in prior. The RDH scheme with block-wise image histograms [22] consumed much time for the encryption procedure and lacked process in real-time applications. The proposed SHA-256 approach outperforms the limitations of these existing methods, the proposed method preserves the integrity of the original data as it only modifies the values of the sensitive data. The proposed method encrypts the masked data for more secure transmission of data.

5. CONCLUSION

Data masking is utilized for securing the data in several organizations from security breaches, malware attacks and unauthorized access. This research proposes an efficient data masking approach for relational databases according to the hash map using the SHA-256 algorithm. The proposed data masking schema is developed to conserve the sensitive data in content format utilizing the masked data and also preserve the masked data size as original data. The proposed method maintains both privacy as well as the confidentiality of the secret features. In every relational database, the non-sensitive or non-original features are selected as concealment attributes. Additionally, the non-sensitive features are chosen as semi-cover features to describe the virtual share columns. Encryption and decryption are also applied after masking the data for highly secure transmitting data. In the future, the proposed method will expand to perform the different real-world applications.




REFERENCES

- [1] L. Meng, L. Liu, X. Wang, and G. Tian, "Reversible data hiding in encrypted images based on IWT and chaotic system," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 16833–16861, May 2022, doi: 10.1007/s11042-022-12415-z.
- [2] F. S. Hassan and A. Gutub, "Novel embedding secrecy within images utilizing an improved interpolation-based reversible data hiding scheme," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 2017–2030, May 2022, doi: 10.1016/j.jksuci.2020.07.008.
- [3] O. P. Singh and A. K. Singh, "Data hiding in encryption–compression domain," *Complex and Intelligent Systems*, vol. 9, no. 3, pp. 2759–2772, Jun. 2023, doi: 10.1007/s40747-021-00309-w.
- [4] Q. Feng, L. Leng, C. C. Chang, J. H. Horng, and M. Wu, "Reversible data hiding in encrypted images with extended parametric binary tree labeling," *Applied Sciences (Switzerland)*, vol. 13, no. 4, pp. 1–15, Feb. 2023, doi: 10.3390/app13042458.
- [5] M. S. Abdalzaher, M. M. Fouda, and M. I. Ibrahim, "Data privacy preservation and security in smart metering systems," *Energies*, vol. 15, no. 19, pp. 1–19, Oct. 2022, doi: 10.3390/en15197419.
- [6] C. H. Yang, C. Y. Weng, and J. Y. Chen, "High-fidelity reversible data hiding in encrypted image based on difference-preserving encryption," *Soft Computing*, vol. 26, no. 4, pp. 1727–1742, Feb. 2022, doi: 10.1007/s00500-022-06745-1.
- [7] S. Panchikkil, S. P. Vegesana, V. M. Manikandan, P. K. Donta, P. K. R. Maddikunta, and T. R. Gadekallu, "An ensemble learning approach for reversible data hiding in encrypted images with fibonacci transform," *Electronics (Switzerland)*, vol. 12, no. 2, pp. 1–20, Jan. 2023, doi: 10.3390/electronics12020450.
- [8] Y. Qiu, Q. Ying, Y. Yang, H. Zeng, S. Li, and Z. Qian, "High-capacity framework for reversible data hiding in encrypted image using pixel prediction and entropy encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 5874–5887, Sep. 2022, doi: 10.1109/TCSVT.2022.3163905.
- [9] S. Kanwal, F. Tao, A. Almogren, A. U. Rehman, R. Taj, and A. Radwan, "A robust data hiding reversible technique for improving the security in e-Health care system," *CMES - Computer Modeling in Engineering and Sciences*, vol. 134, no. 1, pp. 201–219, 2023, doi: 10.32604/cmcs.2022.020255.
- [10] M. Li, Z. Tian, X. Du, X. Yuan, C. Shan, and M. Guizani, "Power normalized cepstral robust features of deep neural networks in a cloud computing data privacy protection scheme," *Neurocomputing*, vol. 518, pp. 165–173, Jan. 2023, doi: 10.1016/j.neucom.2022.11.001.
- [11] Y. Zhang and W. Luo, "Vector-based efficient data hiding in encrypted images via multi-MSB replacement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7359–7372, Nov. 2022, doi: 10.1109/TCSVT.2022.3183391.
- [12] E. Esai Malar and B. Paramasivan, "Enhancing security and privacy preserving of data in cloud using SHA and genetic algorithm," in *Computational Intelligence in Pattern Recognition: Proceedings of CIPR*, Springer Singapore, 2021, pp. 401–411, doi: 10.1007/978-981-16-2543-5_34.




- [13] P. Chen, Z. Zhang, Y. Lei, K. Niu, and X. Yang, "A multi-domain embedding framework for robust reversible data hiding scheme in encrypted videos," *Electronics (Switzerland)*, vol. 11, no. 16, pp. 1–21, Aug. 2022, doi: 10.3390/electronics11162552.
- [14] Y. Y. Tsai, H. L. Liu, P. L. Kuo, and C. S. Chan, "Extending multi-MSB prediction and Huffman coding for reversible data hiding in encrypted HDR images," *IEEE Access*, vol. 10, pp. 49347–49358, 2022, doi: 10.1109/ACCESS.2022.3171578.
- [15] R. Anushiadevi and R. Amirtharajan, "Design and development of reversible data hiding- homomorphic encryption & rhombus pattern prediction approach," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 46269–46292, Dec. 2023, doi: 10.1007/s11042-023-15455-1.
- [16] G. D. Su and C. C. Chang, "Toward high-capacity crypto-domain reversible data hiding with huffman-based lossless image coding," *Visual Computer*, vol. 39, no. 10, pp. 4623–4638, Oct. 2023, doi: 10.1007/s00371-022-02613-z.
- [17] X. Wang, C. C. Chang, C. C. Lin, and C. C. Chang, "Privacy-preserving reversible data hiding based on quad-tree block encoding and integer wavelet transform," *Journal of Visual Communication and Image Representation*, vol. 79, pp. 1–12, Aug. 2021, doi: 10.1016/j.jvcir.2021.103203.
- [18] H. Zhang, L. Li, and Q. Li, "Reversible data hiding in encrypted images based on block-wise multi-predictor," *IEEE Access*, vol. 9, pp. 61943–61954, 2021, doi: 10.1109/ACCESS.2021.3072376.
- [19] P. Rahmani, M. Taheri, and S. M. Fakhrahmad, "A novel secure data outsourcing scheme based on data hiding and secret sharing for relational databases," *IET Communications*, vol. 17, no. 7, pp. 775–789, Apr. 2023, doi: 10.1049/cmu2.12581.
- [20] R. Anushiadevi, P. Praveenkumar, J. B. B. Rayappan, and R. Amirtharajan, "Uncover the cover to recover the hidden secret - A separable reversible data hiding framework," *Multimedia Tools and Applications*, vol. 80, no. 13, pp. 19695–19714, May 2021, doi: 10.1007/s11042-021-10729-y.
- [21] A. Malik, A. Ashraf, H. Wu, and M. Kuribayashi, "Reversible data hiding in encrypted text using Paillier cryptosystem," *Proceedings of 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2022*, pp. 1495–1499, 2022, doi: 10.23919/APSIPAASC55919.2022.9979998.
- [22] K. S. R. Murthy and V. M. Manikandan, "Reversible data hiding using block-wise histogram shifting and run-length encoding," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 74–80, 2021, doi: 10.14569/IJACSA.2021.0120511.
- [23] S. Dhar, A. Khare, and R. Singh, "Advanced security model for multimedia data sharing in internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 11, Nov. 2023, doi: 10.1002/ett.4621.
- [24] B. S. Rawal, S. N. Aleti, and S. Reddy, "Optimization of SHA 256 with finetune pipeline and parallel processing with split techniques," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 3s, pp. 460–472, 2022.
- [25] R. R. Suman, B. Mondal, and T. Mandal, "A secure encryption scheme using a composite logistic sine map (CLSM) and SHA-256," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 27089–27110, Aug. 2022, doi: 10.1007/s11042-021-11460-4.

BIOGRAPHIES OF AUTHORS



Tumkur Shankaregowda Bharath    born and raised in Tumakuru, Karnataka, India. He graduated from Vishweshwaraya Technological University (VTU) and earned M.Tech. from Vishweshwaraya Technological University (VTU). Soon after graduation he started working as a Lecturer at esteemed Institute Sri Siddhartha Institute of Technology (SSIT), Maraluru Tumkur. He has teaching experience of totally 16 years and expertise in various fields like data structures, algorithms, web application development, big data, and cloud computing. He can be contacted at email: bharathts@ssit.edu.in.



Channakrishnaraju    has 27 years of teaching experience for UG and PG courses in computer science and engineering, and is currently working as Professor in the Department of Computer Science and Engineering, in Sri Siddhartha Institute of Technology, Tumkur. He obtained B.E. from Bangalore University in the year 1995 and PG in software systems in the year 2000 From BITS, Pilani and Doctoral degree Ph.D. in computer science and engineering at Sri Siddhartha Academy of higher Education in the year 2017 Tumkur. His research interests are in the areas of wireless sensor networks, network security, artificial intelligence, cloud computing, IoT, architecture, and soft computing. He worked as Editorial member for several international journals and he worked reviewer for many IEEE international conferences. He published more than 47 papers in international journal and conferences and author for book chapter published in Springer publication. He published design patent on "AI and IoT based Solar Bicycle" in the year 2023. He is acting as resource persons for several workshops and conferences. He can be contacted at email: rajuck@ssit.edu.in.